What is claimed is:

1. A processor comprising:

a cache;

an execution unit to execute an instruction having an operand indicating a monitor address;

a bus controller to assert a preventative signal in response to receiving a memory access attempting to gain sufficient ownership of a cache line associated with said monitor address to allow a modification of said cache line without generation of another transaction indicative of the modification.

2. The processor of claim 1 wherein said cache is an L1 cache and wherein said processor further comprises an L2 cache.

3. The processor of claim 2 wherein the cache line associated with said monitor address is flushed from the L1 cache and the L2 cache in response to said instruction.

4. The processor of claim 1 wherein said bus controller is to generate a bus cycle in response to the instruction, the bus cycle to eliminate any ownership of said cache line by another processor that would allow modification of said cache line without generation of another transaction indicative of modification of the cache line.

5. The processor of claim 4 wherein said bus cycle is a read and/or invalidating bus cycle.

6. The processor of claim 5 further comprising a monitor coupled to said bus controller to monitor bus transactions for a transaction indicative of a write to the monitor address, and to signal a monitor event in response to the transaction indicative of the write to the monitor address.

7. The processor of claim 2 further comprising:

a plurality of write combining buffers between said L1 cache and said L2 cache;

a snoop port for said plurality of write combining buffers;

a monitor coupled to said L1 cache and coupled to the snoop port to monitor memory access cycles from the execution unit and from the snoop port.

8. An apparatus comprising:

a bus controller having a plurality of bus cycle information lines;

programmable memory access detection logic coupled to a bus, said programmable memory access detection logic comprising a storage location to store a monitor address specified by an instruction and comparison logic having inputs coupled to said storage location and said plurality of bus cycle information lines and a comparison logic output;

coherence logic coupled to receive said monitor address, that in response to the instruction is to generate a read and/or invalidate transaction for a cache line associated with said monitor address.

9. The apparatus of claim 8 wherein said programmable memory access detection logic

comprises write detection logic.

10. The apparatus of claim 9 further comprising:

hit generation logic coupled to said storage location and said bus controller, wherein said hit generation logic has an output hit signal externally available to couple to a system bus.

11. The apparatus of claim 8 wherein said bus controller is to generate, in response to the instruction, a bus cycle chosen from a set consisting of:

a bus read line invalidate of the cache line associated with said monitor address;

a bus write line invalidate of the cache line associated with said monitor address.

12. The apparatus of claim 8 wherein said read and/or invalidate transaction is to ensure that no other processor caches include said cache line associated with said monitor address in a modified or exclusive state.

13. The apparatus of claim 8 wherein said instruction is a part of a first thread, further comprising:

suspend logic responsive to a second instruction in said first thread to suspend said first thread;

resume logic responsive to said comparison logic output to resume said first thread.

14. The apparatus of claim 13 further comprising:

partition/anneal logic to anneal and partition resources responsive to respectively suspension and resumption of said first thread.

15. A method comprising:

performing a first bus transaction to eliminate ownership by other agents of a cache line associated with a monitor address specified by an instruction;

asserting a preventative signal in response to a second bus transaction attempting to gain ownership of said cache line associated with the monitor address.

16. The method of claim 15 wherein performing the first bus transaction comprises:

preventing any system processor cache from storing said cache line associated with said monitor address in a modified or exclusive state.

17. The method of claim 15 wherein performing the first bus transaction comprises performing an invalidating transaction.

18. The method of claim 15 wherein performing the first bus transaction comprises performing a read transaction.

19. The method of claim 15 wherein asserting the preventative signal comprises asserting a hit signal in response to a transaction which could result in a bus agent gaining ownership of the cache line associated with said monitor address.

20. The method of claim 15 wherein said monitor address is an operand of said instruction.

21. The method of claim 15 further comprising flushing said cache line from a plurality of processor caches in a processor that executes said instruction.

22. The method of claim 15 further comprising:

suspending execution of a first thread of which the instruction is a part, in response to a second instruction;

resuming execution of said first thread in response to detection of a memory access to the monitor address.

23. The method of claim 22 wherein suspending execution of the first thread further comprises:

relinquishing a plurality of thread partitionable resources associated with said first thread.

24. A system comprising:

a bus;

a first processor having a first cache;

a second processor having a second cache, said second processor comprising:

a monitor to monitor transactions from the first processor on the bus to detect a memory access to a monitor address specified by an instruction executed by said second processor;

coherence logic to generate a bus transaction to prevent said first cache from owning a cache line associated with said monitor address.

25. The system of claim 24 wherein said second processor further comprises:

hit generation logic to generate a hit signal in response to a read transaction to said cache line associated with said monitor address.

26. The system of claim 24 wherein said second processor is to flush said cache line associated with said monitor address from the second cache in response to said instruction.

27. The system of claim 24 wherein said bus transaction to prevent said first cache from owning said cache line associated with said monitor address is a read transaction.

28. The system of claim 24 wherein said bus transaction prevents the first cache from holding said cache line associated with said monitor address in a modified or exclusive state.

29. The system of claim 24 wherein said second processor further comprises thread suspension logic to suspend a first thread of which said instruction is a part until an access to said cache line associated with said monitor address occurs.

30. The system of claim 29 wherein said second processor further comprises partitioning and annealing logic to relinquish resources associated with said first thread when said first thread is suspended and to re-partition resources to accommodate said first thread when said first thread is resumed.